Exercise 1: Full Adder

A full adder is a combinational circuit that adds three bits (A, B, and the carry-in, Cin) and generates a sum (S) and a carry-out (Cout).

1. Construct the truth table for the full adder. List all possible input combinations (A, B, Cin) and calculate the sum (S) and the carry-out (Cout).
2. Express the sum (S) and the carry-out (Cout) in terms of the inputs A, B, and Cin.
3. Simplify the logical expressions for the sum (S) and the carry-out (Cout) using Boolean algebra laws.
4. Provide the logic diagram of the full adder using AND, OR, and XOR gates.
5. Provide the logic diagram of the full adder in terms of the half adder.

Exercise 2: Full Subtractor

A full subtractor is a circuit that takes three inputs (A, B, and the borrow-in, Bin) and generates a difference (D) and a borrow-out (Bout).

1. Construct the truth table for the full subtractor. List all possible input combinations (A, B, Bin) and calculate the difference (D) and the borrow-out (Bout).
2. Express the difference (D) and the borrow-out (Bout) in terms of the inputs A, B, and Bin.
3. Simplify the logical expressions for D and Bout.
4. Provide the logic diagram of the full subtractor using AND, OR, and XOR gates.
5. Provide the logic diagram of the full adder in terms of the half substractor.

Exercise 3: 2-Bit Multiplier

A 2-bit multiplier takes two 2-bits binary numbers (A = A1A0 and B = B1B0) and generates a 4-bits product (P = P3P2P1P0).

1. Construct the truth table for the 2-bit multiplier. List all possible input combinations for A and B and calculate the product P = P3P2P1P0
2. Express the product bits (P3, P2, P1, P0) in terms of the input bits A and B.
3. Write the logical equations for each bit of the product.
4. Simplify the logical expressions for the product bits, if possible.
5. Provide the logic diagram of the 2-bit multiplier.

Exercise 4: Full Subtractor Using Full Adder

You need to design a full subtractor using a full adder. Remember that a full subtractor is a circuit that takes three inputs: a borrow-in bit (Bin), and two bits to subtract (A and B), and produces a difference (D) and a borrow-out (Bout). However, instead of designing a full subtractor directly from logic gates, the idea here is to reuse the structure of a full adder to perform subtraction.

1. Provide the logic diagram of the full subtractor using a full adder.
2. Verify the truth table of the full subtractor to ensure that the outputs (D and Bout) are correct for each combination of the inputs A, B, and Bin.

Exercise 5: Design of a Multiplier Using a Full Adder

You need to design a 2-bit multiplier using full adders. This multiplier will take two 2-bit binary numbers (A = A1A0 and B = B1B0) and produce a 4-bit product (P = P3P2P1P0). The goal is to implement this multiplier using full adders to add the partial products of the bits.

1. Provide the logic diagram of the multiplier using AND gates and full adders to add the partial products.

Exercise 6: Realize a Full Adder and a Full Subtractor in the Same Circuit

You need to design a combinational circuit capable of performing both a full adder and a full subtractor depending on a control signal. This signal will determine whether the operation to be performed is an addition or a subtraction.

- **Inputs:**
  - A, B: The two bits to add or subtract.
  - Cin: The carry-in or borrow-in (for both addition and subtraction).
  - Mode: A control bit that determines the operation (0 for addition, 1 for subtraction).
- **Outputs:**
  - S: The sum or difference, depending on the operation selected.
  - Cout / Bout: The carry-out for addition or the borrow-out for subtraction.

Design a combinational circuit where both a full adder and a full subtractor are implemented in parallel, but the behavior of the circuit is controlled by the **Mode** signal.

- When **Mode = 0**, the circuit should behave as a full adder.
- When **Mode = 1**, the circuit should behave as a full subtractor.
- **Truth Table:** Construct the truth table for the circuit. List all possible combinations of inputs (A, B, Cin, Mode) and determine the output values (S, Cout/Bout).